# UDP/IP Integration Software Testbench Instructions

Team Gamma

April 16, 2017

## Dependencies

- Xilinx Software Development Kit v2016.4
- 1 Digilent ZedBoard
- `integration_test.sdk.zip`

For the duration of this document, let the workspace directory created in the procedure equal `WKSPACE`.

## Test Inputs

| Filename | Description | Test |
|---|---|---|
| `WKSPACE/integration_test/src/rx_test_data.h` | UDP/IP receiver integration and throughput tests | 24, 26 |
| `WKSPACE/integration_test/src/tx_test_data.h` | UDP/IP transmitter integration and throughput tests | 25, 27 |

Note that these inputs are the same as those for the HDL testbench.

## Procedure

1. Launch the Xilinx SDK, selecting an empty folder for the new workspace. If no prompt to select a workspace appears, select `File > Switch Workspace > Other...` and select an empty folder for the new workspace.
2. Select `File > Import`.
3. Select `General > Existing Projects into Workspace`.
4. Click the radio button next to `Select archive file:` and then use `Browse` to select `integration_test.sdk.zip`.
5. All projects already should be selected, click `Finish`.
6. If build errors were encountered, select `Project > Clean`, select `Clean all projects`, then click `OK`. All projects should automatically rebuild successfully after cleaning.
7. If the ZedBoard does not have a boot image installed where boot can be stopped at U-Boot or earlier, set the boot mode jumpers to JTAG boot (refer to the ZedBoard Hardware User's Guide, section 2.10).
8. Power on the ZedBoard and connect both the serial and JTAG USB connections to the workstation.
9. Connect to the ZedBoard's serial connection.

10. Program the bitstream by selecting `Xilinx Tools > Program FPGA`. `design_1_wrapper.bit` should already be selected, click `OK`.
11. In the `Project Explorer`, right-click `integration_test > Binaries > integration_test.elf` and select `Run As > Launch on Hardware (System Debugger)`.
12. Output on the serial connection should be seen as in appendix B.

## Notes

- A 64 bit interface was not used on the AXI memory-mapped FIFO since the driver had difficulty running a pure loopback with it. Instead the FIFO is 32 bits wide, and width converters are used between the UDP/IP modules and the memory-mapped FIFO's streaming interfaces.
- Transmission integration tests 2 and 3 are disabled because of what is currently believed to be an alignment issue with the FIFO output to the modules.

## A External Software Links

- Xilinx Software Development Kit

## B Expected Output

```
Entering UDP/IP integration test suite

START: RX Integration Test 1
send: Waiting for transmission to end...
recv: Waiting for input frame...
recv: Frame length 12, requested 14 bytes.
recv: Read 12 bytes.
PASS

START: RX Integration Test 2
send: Waiting for transmission to end...
recv: Waiting for input frame...
recv: Frame length 12, requested 15 bytes.
recv: Read 12 bytes.
PASS

START: RX Integration Test 3
send: Waiting for transmission to end...
recv: Waiting for input frame...
recv: Frame length 16, requested 17 bytes.
recv: Read 16 bytes.
PASS

START: RX Throughput Test 1
send: Waiting for transmission to end...
recv: Waiting for input frame...
recv: Frame length 1408, requested 1412 bytes.
recv: Read 1408 bytes.
PASS
```

```
START: RX Throughput Test 2
send: Waiting for transmission to end...
recv: Waiting for input frame...
recv: Frame length 1408, requested 1412 bytes.
recv: Read 1408 bytes.
PASS

START: TX Integration Test 1
send: Waiting for transmission to end...
recv: Waiting for input frame...
recv: Frame length 40, requested 44 bytes.
recv: Read 40 bytes.
PASS

START: TX Throughput Test 1
send: Waiting for transmission to end...
recv: Waiting for input frame...
recv: Frame length 1428, requested 1432 bytes.
recv: Read 1428 bytes.
PASS

START: TX Throughput Test 2
send: Waiting for transmission to end...
recv: Waiting for input frame...
recv: Frame length 1428, requested 1432 bytes.
recv: Read 1428 bytes.
PASS

Exiting...
```